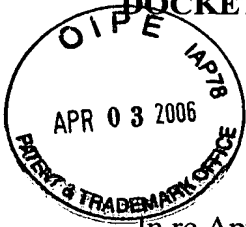


04-05-06

APR
9/11

DOCKET NO.: MSFT-0764/154583.01

PATENT



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: **Richard Ciapala,
et al.**

Confirmation No.: **2868**

Serial No.: **10/053,376**

Group Art Unit: **2194**

Filing Date: **November 9, 2001**

Examiner: **Anya, Charles E.**

For: **TASK TRACING IN A DISTRIBUTED COMPUTING ENVIRONMENT**

EXPRESS MAIL LABEL NO: EV 765641679 US
DATE OF DEPOSIT: April 3, 2006

Mail Stop Appeal-Brief Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

APPELLANT'S BRIEF PURSUANT TO 37 C.F.R. § 41.37

This brief is being filed in support of Appellant's appeal from the final rejections of claims 1-11, and 23-26 dated June 21, 2005. A Notice of Appeal was filed on December 19, 2005.

04/06/2006 TBESHAH1 00000014 10053376

01 FC:1402

500.00 OP

1. REAL PARTY IN INTEREST

The real party in interest is Microsoft Corporation of Redmond, Washington. This case has been assigned to Microsoft Corporation, as evidenced by the assignment recorded on April 1, 2002 at Reel 012786, Frame 0176.

2. RELATED APPEALS AND INTERFERENCES

There are presently no related appeals or interferences.

3. STATUS OF CLAIMS

Claims 1-11 and 23-26 are pending in this case, all of which stand finally rejected as a result of the Final Office Action dated June 21, 2005. This appeal is from the rejection of claims 1-11 and 23-26.

4. STATUS OF AMENDMENTS

Claim 10 was amended on August 22, 2005, after the Final Office Action from which this appeal is taken. By Advisory Action dated September 20, 2005, the Examiner stated that the amendment would be entered for the purpose of the appeal. In the claims appendix, the text of claim 10 reflects the status of that claim following the after-final amendment.

5. SUMMARY OF CLAIMED SUBJECT MATTER

This case is about tools and methods that are used to analyze and debug software.

By way of background, analysis and debugging of software normally involves allowing a programmer or systems analyst to observe the internal operation of the software as it runs. This observation can be enabled by having the software itself provide information about “events” that occur during the operation of the software. The programmer or analyst can observe the events to determine how the program is behaving. Based on the observed events, the programmer or analyst can modify the program (or modify some other variable, such as input to the program) in order to remove bugs, increase performance, or achieve some other goal. The invention described and claimed is not limited to any particular modification of a program, or any particular action that a programmer or analyst might take based on observed behavior of the program’s behavior. Rather, the claims are directed to methods and systems that control what information is provided and/or how the information is provided, which helps the programmer or analyst obtain more useful information about the program than might otherwise be received. Certain conventional systems allow programs to generate event information that can be used by programmers and analysts, but in many cases such information is generated without regard to what kind of information is needed or useful. The subject matter of the claims on appeal addresses this issue, as well as others.

With this background having been explained, appellants turn to the claims that are the subject of this appeal.

The claims are generally directed to methods, systems, and computer-readable media that allow event information to be generated by a program in accordance with some instruction or limitation. As to the independent claims:

Claim 1 is directed to “a method for tracing a computing task in a distributed computing environment.” Two devices – a first device and a second device – are referred to in the claim. (See page 22, line 16 through page 23, line 10; and elements 604, 606 in FIG. 6.) The first device issues a call to invoke a procedure that is to be executed at the second device, and the call includes tracing information “instructing” the second device to provide event information regarding the execution of the procedure. (See page 22, line 16 through page 23, line 10; page 14, line 3 et seq.; page 15, lines 16-28; elements 404, 406 in FIG. 4.) When the procedure is invoked at the second device, event information is provided in accordance with the tracing information – i.e., in accordance with what the tracing

information has instructed. (See page 14, line 3 et seq.; page 15, lines 16-28; elements 404, 406, 422 in FIG. 4.)

Claim 10 is directed to a computer-readable medium having instructions to perform certain acts related to the generation of event information. The acts recited in claim 10 call for certain event information to be generated; certain event information is generated during the operation of a procedure on a remote device. (See page 11, line 27 through page 12, line 10; page 22, line 16 through page 23, line 10; elements 304(1), 304(2), 304(n) in FIG. 3; elements 604, 606 in FIG. 6) In particular, claim 10 calls for: “calling a procedure on a remote device whose location or identity is undetermined at the time of the call.” (See page 22, lines 12-15.)

Claim 23 is directed to a “system for supporting tracing in an application program.” The application program executes on a first computing device and issues a call to a second computing device. (See page 22, line 16 through page 23, line 10; and elements 604, 606 in FIG. 6.) The system comprises a library (see page 16, lines 21-25; element 502, FIG. 5) that comprises one or more methods callable by the application program. Calls to the methods generate events, and an event handler causes the generation of tracing information in response to the events. (See page 17, line 21 through page 18, line 18.) Moreover, the generation of tracing information is “limited by a requirement that originates from the application program.” (See page 12, line 10 through page 13, line 12.)

Claim 25 is directed to a computer-readable medium having executable components for “supporting tracing in an application program.” The application program executes on a first computing device and issues a call to a second computing device. (See page 22, line 16 through page 23, line 10; and elements 604, 606 in FIG. 6.) While claim 25 is not identical to claim 23 in either language or scope, for the sake of simplifying the issues before the Board, appellants direct the Board’s attention to certain features of claim 25 that are similar to those described above for claim 23. In particular, claim 25 defines a library that comprises one or more methods callable by the application program, and an event handler causes the generation of tracing information in response to the events. (See page 17, line 21 through page 18, line 18.) Moreover, the generation of tracing information is “limited by a limitation requirement that is sent from the first computing device to the second computing device.” (See page 12, line 10 through page 13, line 12.)

Appellants have summarized the general topic of this application and the independent claims thereof in order to provide an overview of the subject matter involved in this appeal.

However, each claim, whether independent or dependent, should be considered on its own merits in accordance with the arguments made below in section 7.

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1-7, 9, and 23-26 are anticipated by U.S. Patent No. 6,760,903 (Morshed) under 35 U.S.C. § 102(e).

2. Whether claims 8, 10, and 11 are obvious over Morshed in view of U.S. Patent No. 6,46,137 (Vasudevan) under 35 U.S.C. § 103(a).

7. ARGUMENT

1. Claims 1-7, 9, and 23-26 are not anticipated by Morshed

Appellants submit that claims 1-7, 9, and 23-26 are not anticipated by Morshed under 35 U.S.C. § 102(e) because: (1) Morshed does not teach or suggest that a first device instructs a second device to provide event information about a procedure that executes at a first device; and (2) Morshed does not teach or suggest that a limitation on the content of event or tracing information is specified and honored.

a. **Morshed does not teach or suggest that a first device instructs a second device to provide event information about a procedure that executes at a first device (claim 1).**

Independent claim 1 calls for a first device to issue a first call that includes tracing information “instructing [a] second device to provide information regarding the execution of [a] first procedure at the second device.” Morshed does not teach or suggest this feature.

As to this feature, the Examiner applied col. 35, ll. 18-31 of Morshed. The applied portion of Morshed is reproduced below for the Board’s convenience:

With regard to the occurrence of the first event in connection with an outgoing call from a client system to a server system, the hook specified as point of transfer in the monitor DLL inserts “out of band data” into the stream of data that is transferred to the server. In this particular embodiment, this additional data or “out of band data” is used to pass information about the calling function included in the client process 1030. An example of information that may be included in this call as out of band data is described elsewhere herein. This information may be subsequently extracted on the server side and may be used in gathering execution data about the distributed application as a whole as well as components of the distributed application that may execute on different computer systems.

The applied portion of Morshed generally explains that “execution data” may be gathered about the execution of a called function. The applied portion further explains that “out of band” data transmitted with the function call may be used in the gathering of the execution data. (The premise of the Examiner’s position is that Morshed’s “execution data” corresponds to the claimed “event information.”)

The applied portion of Morshed does not show that one device instructs another to provide the “execution information.” Thus, even if the “execution data” that is gathered in Morshed can be considered analogous to the “event information” recited in claim 1, claim 1 is distinguishable from Morshed on the ground that Morshed does not *instruct* a second device to provide the execution information.

During prosecution of this case before the Examiner, the Examiner advanced two explanations as to how Morshed teaches that one device instructs another to provide event information. In essence, the Examiner's position is that either: (a) the out of band data contains an instruction to provide execution data; or (b) a call from a client to a server (as described in the above-quoted portion) is itself an instruction to provide execution data.¹ For the reasons explained below, Morshed does not show either (a) or (b).

First, as to point (a), Morshed does not show that the "out of band" data contains an instruction to provide the event information recited in claim 1 (or the "execution data" referred to in Morshed). None of the various references to "out of band" data in Morshed show that "out of band" data is used to instruct that a device provide execution data. No such instruction is shown in the above-quoted portion of Morshed; nor have appellants been able to identify any other portion of Morshed that describes "out of band" data as containing an instruction to provide execution information. While the above-quoted portion of Morshed does indicate that the "out of band" data may be "used in gathering execution data about the distributed application," this fact does not demonstrate that the out of band data *instructs* a device to provide execution data.

The fact that out of band data is "used" to gather execution data does not mean that the out of band data *instructs* a device to provide event information. In fact, Morshed describes what type of information is contained in the "out of band" data, and that information is *not* an instruction to provide event information (or execution data). Specifically, Morshed characterizes the "out of band" data in the following passage: "... out of band data, such as information about the calling function sent from the client system." (Morshed, col. 39, ll. 9-10.) "[I]nformation about the calling function" is not an instruction to provide event information. As noted above, claim 1 calls for the first device to issue a call to a procedure at a second device that, among other things, instructs the second device to provide event information. The Examiner's position is essentially that the client system from which the call is made corresponds to the claimed "first device," and the server system to which the call is issued corresponds to the claimed "second device." Under this analogy, the "information about the calling function" that may be part of the "out of band" data would have to instruct the server to provide event information (or execution data). However, a

¹ These arguments were raised by the Examiner in an oral interview held on July 12, 2005, and are summarized in appellant's paper responding to the June 21, 2005 Office Action/Final Rejection. While these arguments are not part of the written Final Rejection from which this appeal has been taken, appellants have discussed these arguments so that the Board can fully understand and address the issues that were argued below.

caller's sending information about itself to the callee does not *instruct* the callee to do anything.

Thus, it can be seen that Morshed's "out of band" data does not *instruct* a device to provide event information, as called for in claim 1.

Turning now to point (b), the call from a client to a server is not an instruction to provide event information. While a call from client to server is arguably an "instruction" to *execute a function on the server*, such a call does not instruct the server to *provide event information*. Clearly, it is possible to call a function on the server without instructing the callee to provide event information. In Morshed, the call from the client to the server causes the server to perform whatever action the client is requesting of the server; there is no suggestion in the above-quoted portion of Morshed, or in any other portion, that the call instructs the server to provide event information (or execution data). In Morshed, it appears that execution data will be provided irrespective of what information might be contained in the call from client to server. Thus, the call cannot be understood as instructing a device to provide event information, as recited in claim 1.

For these reasons, appellants submit that Morshed does not teach every limitation of claim 1, and thus does not anticipate claim 1. Appellants request that the Board reverse the rejection of claim 1.

b. Morshed does not teach or suggest that a limitation on the content of event information is specified and honored (claim 2)

Claim 2 (dependent on claim 1) states that "tracing information" (which, as defined in claim 1, is included in the call issued at the first device) "specifies a limitation on the content of the event information, and wherein said act of providing event information comprises providing a limited amount of event information in accordance with the specified limitation." Morshed does not teach or suggest this feature.

In rejecting claim 2, the Examiner relied on col. 35, ll. 18-31 of Morshed, which is quoted above in the section discussing claim 1. In particular, the Examiner asserted that that the "out of band" data specifies the "limitation" called for in claim 2.

Morshed's "out of band" data does not teach or suggest that a "limitation" on the content of event information to be provided. As discussed above, the "out of band" data may be "used" in gathering execution information. However, the fact that out of band data may be "used" to gather execution information does not mean that the out of band data specifies a *limitation* on the data to be gathered. Neither the applied portion of Morshed, nor any other

portion that appellants have been able to identify, indicates that the out of band data imposes a limitation on what event information is to be provided. As noted above, Morshed indicates that the out of band data contains information about the caller; however, information about the caller is not the same as a limitation on the event information to be provided.

Claim 2 is patentable by virtue of its dependency on claim 1. Additionally, for the reasons stated above, Morshed does not teach the limitations recited in claim 2, and thus does not anticipate claim 2. Appellants request that the Board reverse the rejection of claim 2.

c. Morshed does not teach or suggest that an application program originates a requirement that limits the generation of tracing information (claim 23)

Claim 23 recites “an event handler residing on [a] first computing device.” The event handler “receives events” and “causes the generation of first tracing information in response to said events.” Moreover, “the generation of said first tracing information [is] limited by a requirement that originates from the application program.” In other words, the application program imposes a requirement that ultimately limits that tracing information that is generated. Claim 23 stands rejected as being anticipated by Morshed; specifically, the Examiner has cited Morshed’s discussion of “out of band” data (col. 35, ll. 18-31, quoted above in section 7.1.a) as teaching the feature that the tracing information is limited by a requirement imposed by the application program. As discussed above, Morshed’s “out of band” data does not impose a limitation on what information is created; it is merely “used” in the gathering of information.

For these reasons, appellants submit that Morshed does not teach every limitation of claim 23, and thus does not anticipate claim 23. Appellants request that the Board reverse the rejection of claim 23.

d. Morshed does not teach or suggest that a first computing device sends a requirement that limits the generation of tracing information (claim 25)

Claim 25 recites an “event handler” that receives events and causes “the generation of first tracing information in response to said events.” Additionally, claim 25 recites that “the generation of said first tracing information [is] limited by a limitation requirement that is sent from [a] first computing device to [a] second computing device.” While claim 25 is not identical to claim 23 in either language or scope, we note that the above-quoted features are similar to the features discussed above in connection with claim 23.

The Examiner below did not set forth an explanation as to why claim 25 should be rejected over Morshed. Instead the Examiner simply referred to the rejection of claim 23 as providing the rationale. As noted above, however, Morshed's "out of band" data does not teach or suggest the feature that the tracing information is limited by a requirement imposed by the application program, as in claim 23. For essentially the same reason, it is clear that Morshed's "out of band" data does not teach or suggest the feature of limiting tracing information by a requirement sent from a first device to a second device. As previously noted, Morshed's "out of band" data is not a limitation on the generation of tracing information.

For these reasons, appellants submit that Morshed does not teach every limitation of claim 23, and thus does not anticipate claim 23. Appellants request that the Board reverse the rejection of claim 23.

e. Claims 3-7, 9, 24, and 26 are not anticipated by Morshed

Claims 3-7, 9, 24, and 26 are dependent on claims that have been shown to be patentable, and, therefore, are patentable at least by reason of dependency.

2. Claims 8, 10, and 11 are not obvious over Morshed in view of Vasudevan

Appellants submit that claims 8, 10, and 11 are not obvious over Morshed in view of Vasudevan under 35 U.S.C. § 103(a). At a minimum, claim 10 is not obvious over this combination because these references do not teach or suggest that a procedure is called whose identity or location is undetermined at the time of the call, as recited in claim 10. Additionally, claims 8 and 11 are dependent on claims that have been shown to be patentable, and are therefore also patentable.

a. Morshed and Vasudevan do not teach or suggest that a procedure is called whose identity or location is undetermined at the time of the call

Claim 10 recites the feature of "calling a procedure on a remote device whose location or identity is undetermined at the time of the call." The Examiner below rejected claim 10 as being unpatentable over a proposed combination of Morshed and Vasudevan. However, it should be emphasized that the Examiner has never asserted that this feature is present in Morshed; in fact, the Examiner admits that this feature is not present in Morshed. In the Examiner's words:

Morshed is silent with reference to calling a procedure whose location or identity is undetermined at the time of the call. [Final Rejection, paragraph 23.]

Instead, the Examiner relied entirely on Vasudevan as teaching the above-quoted feature. Thus, with respect to claim 10, the issue on appeal is whether the above-quoted feature is present in Vasudevan.

Vasudevan does not teach or suggest the above-quoted feature of claim 10. Vasudevan explains that the identity of a called procedure might be unresolved at *compile time*. However, *compile time* is prior to *the time of the call*, since the program is compiled, and then later started, before any procedures can be called. A program in Vasudevan runs an initialization procedure that assigns a “back end” to handle function calls, so by the time a function is called the identity of the back end that will handle function calls for the program is known. In particular, column 12 of Vasudevan explains as follows:

The application 116 includes a call to initialize the handle in the client stub 120 to the VRPC backend 140 used to invoke the backend's send_call method. The application 116 invokes 400 this initialization method, for example VRPC_begin(), on the VRPC library 122, passing in an identification of a server interface being accessed. ... At some subsequent point, the main routine 118 invokes 409 the client stub 120 by calling one of the procedures defined therein, and passing in some number of arguments. The client stub 120 packages the actual arguments and the argument specification to create the procedure call, specifying the argument mode(s) and type of each argument, and the procedure description, specifying the procedure number, number of arguments, return type. The client stub 120 then invokes 408 the send_call method. This call is forwarded by the VRPC library 122 to the selected VRPC engine 144 on the client computer 110.

In other words, the identity of the server that will handle the call is determined as part of an initialization procedure performed at runtime, and by the time an actual procedure call is made, the identity of this server is known. By contrast, in claim 10, the identity or location of the procedure being called is undetermined at the time of the call. Thus, Vasudevan does not teach the feature for which it is cited.

It is important to note that the Examiner's rejection of claim 10 rests on the argument that Vasudevan teaches calling a procedure whose identity or location is undetermined at the time of the call. The Examiner has never proposed any other theory based on modifications or combinations of Morshed or Vasudevan that would yield this claim feature. However, appellants have reviewed both the Morshed and Vasudevan applications, and do not see any basis to find that the features of claim 10 can be derived from either a combination or reasonable modification of Morshed and Vasudevan.

Accordingly, appellants submit that claim 10 is not obvious over Morshed and Vasudevan. Appellants thus request that the rejection of claim 10 be reversed.

b. Claims 8 and 11 are not obvious over Morshed in view of Vasudevan

Claims 8 and 11 are dependent, respectively, on claims 1 and 10. Since claims 1 and 10 have been shown to be patentable, claims 8 and 11 are patentable at least by reason of dependency.

3. Conclusion of Argument

For the foregoing reasons, appellants respectfully submit that the claims on appeal are patentable over the art that was applied below. Applicants thus request that the Board reverse the rejection of the claims on appeal.

8. CLAIMS APPENDIX

1. A method for tracing a computing task in a distributed computing environment, comprising:

at a first device, issuing a first call to invoke a first procedure to be executed at a second device that is different from said first device, said first call including tracing information instructing said second device to provide event information regarding the execution of said first procedure at the second device;

at said second device, receiving the first call and invoking the first procedure in response to said first call; and

at said second device, providing event information in accordance with said tracing information.

2. The method of claim 1, wherein said tracing information specifies a limitation on the content of the event information, and wherein said act of providing event information comprises providing a limited amount of event information in accordance with the specified limitation.

3. The method of claim 1, wherein said event information includes property information descriptive of the event, and wherein said act of providing said event information includes providing said property information.

4. The method of claim 3, further comprising the act of deriving at least some of said property information from an environment present at said second device.

5. The method of claim 3, wherein said property information includes a plurality of attributes, wherein said tracing information specifies a limitation as to a subset of said attributes, and wherein said act of providing event information includes providing attributed information limited in accordance with said subset.

6. The method of claim 1, wherein said first procedure produces a result, and wherein said method further comprises providing said result to said first device.

7. The method of claim 1, wherein said first procedure issues a second call to invoke a second procedure at a third device different from said first device and said second device, and wherein said method further comprises including said tracing information, or information based on said tracing information, in said second call.

8. The method of claim 1, wherein said second device is a member of a cluster of devices, and wherein said first call is issued to said cluster of devices and assigned to said second device, the identity of said second device being indeterminate at the time of said first call.

9. The method of claim 1, further comprising formatting said event information in accordance with a formatting convention.

10. A computer-readable medium having computer-executable instructions to perform acts comprising:

- determining that generation of event information is enabled;
- generating first event information indicative of a first event occurring during the operation of a program;
- calling a procedure on a remote device whose location or identity is undetermined at the time of the call; and
- transmitting to said remote device information instructing said remote device to generate second event information indicative of a second event occurring during the operation of said procedure,

wherein said second event information comprises a plurality of elements, wherein said transmitting act includes transmitting filtering information which limits the second event information to be generated to a subset of said plurality of elements.

11. The computer-readable medium of claim 10, wherein said generating act includes generating property information descriptive of said first event.

23. A system for supporting tracing in an application program which executes on a first computing device and which issues a call to a second computing device for at least some processing, the system comprising:

a library residing on the first computing device comprising one or more methods callable by the application program;

an event handler residing on the first computing device which receives events generated by calls to said methods, and which causes the generation of first tracing information in response to said events, the generation of said first tracing information being limited by a requirement that originates from the application program; and

a trace service component which receives at least some of said tracing information and which generates a remote trace request for forwarding to the second computing device when said tracing information indicates that the application program has issued a call to the second computing device.

24. The system of claim 23, wherein the call to the second computing device is represented in the form of a data structure to be transmitted to the second computing device over a communications medium, and wherein said trace service component attaches the remote trace request to said data structure.

25. A computer-readable medium having stored thereon a plurality of computer-executable components for supporting tracing in an application program that executes on a first computing device and that issues a call to a second computing device for at least some processing, the components comprising:

a library which is installable on the first computing device, said library comprising one or more methods that are executable on the first computing device and that are callable by the application program;

an event handler which is installable and executable on the second computing device, said event handler receiving events generated by calls to said methods and causing the generation of first tracing information in response to said events, the generation of said first tracing information being limited by a limitation requirement that is sent from the first computing device to the second computing device; and

a trace service component which is executable on the first computing device, which receives at least some of said tracing information, and which generates a remote trace request for forwarding to the second computing device when said tracing information indicates that the application program has issued a call to the second computing device.

26. The computer-readable medium of claim 25, wherein the call to the second computing device is represented in the form of a data structure to be transmitted to the second computing device over a communications medium, and wherein said trace service component attaches the remote trace request to said data structure.

9. EVIDENCE APPENDIX

There is no additional evidence to be submitted in this Appeal.

10. RELATED PROCEEDINGS APPENDIX

No related appeals or interferences are pending. Therefore, Appellant is not providing copies of any related proceedings.

Respectfully submitted,

A handwritten signature in black ink, appearing to be 'P. M. Ullman', written over a horizontal line.

Date: April 3, 2006

Peter M. Ullman
Registration No. 43,963

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439